



Rocky Enterprise Linux 9.2 Manual Pages on command 'Regex::Pattern::License.3pm'

C:\>man Regex::Pattern::License.3pm

Regex::Pattern::License(3pmUser Contributed Perl DocumentatiRegex::Pattern::License(3pm)

NAME

Regex::Pattern::License - Regular expressions for legal licenses

VERSION

Version v3.2.0

DESCRIPTION

Regex::Pattern::License provides a hash of regular expression patterns related to legal software licenses.

Regex::Pattern is a convention for organizing reusable regex patterns.

PATTERNS

Single licenses

Patterns each covering a single license.

Each of these patterns has exactly one of these tags: type:singleversion:*
type:unversioned type:versioned:decimal .

? adobe_glyph

? afl

? afl_1_1

? afl_1_2

? afl_2

? afl_2_1

? afl_3

? agpl

? agpl_1

? agpl_2

? agpl_3

? aladdin

? aladdin_8

? aladdin_9

? apache

? apache_1

- ? apache_1_1
- ? apache_2
- ? apafml
- ? artistic
- ? artistic_1
- ? artistic_1_cl8
- ? artistic_1_perl
- ? artistic_2
- ? bdwgc
- ? bdwgc_matlab
- ? beerware
- ? bsd_2_clause
- ? bsd_3_clause
- ? bsd_4_clause
- ? bsl
- ? bsl_1
- ? cc_by

? cc_by_1

? cc_by_2

? cc_by_2_5

? cc_by_3

? cc_by_4

? cc_by_nc

? cc_by_nc_1

? cc_by_nc_2

? cc_by_nc_2_5

? cc_by_nc_3

? cc_by_nc_4

? cc_by_nc_nd

? cc_by_nc_nd_1

? cc_by_nc_nd_2

? cc_by_nc_nd_2_5

? cc_by_nc_nd_3

? cc_by_nc_nd_4

? cc_by_nc_sa

? cc_by_nc_sa_1

? cc_by_nc_sa_2

? cc_by_nc_sa_2_5

? cc_by_nc_sa_3

? cc_by_nc_sa_4

? cc_by_nd

? cc_by_nd_1

? cc_by_nd_2

? cc_by_nd_2_5

? cc_by_nd_3

? cc_by_nd_4

? cc_by_sa

? cc_by_sa_1

? cc_by_sa_2

? cc_by_sa_2_5

? cc_by_sa_3

? cc_by_sa_4

? cc_cc0

? cc_cc0_1

? cc_nc

? cc_nc_1

? cc_nd

? cc_nd_1

? cc_sa

? cc_sa_1

? cc_sp

? cddl

? cddl_1

? cddl_1_1

? cecill

? cecill_1

? cecill_1_1

? cecill_2

? cecill_2_1

? cecill_b

? cecill_b_1

? cecill_c

? cecill_c_1

? cpl

? cpl_1

? cube

? curl

? dsdp

? epl

? epl_1

? epl_2

? eurosym

? fsfap

? fsful

? fsfullr

? ftl

? gfdl

? gfdl_niv

? gpl

? isc

? icu

? ipl

? ipl_1

? json

? jython

? kevin_henney

? lgpl

? lgpl_bdwgc

? libpng

? lgpl

? mit_advertising

? mit_cmu

? mit_cmu_warranty

? mit_enna

? mit_feh

? mit_new

? mit_new_materials

? mit_old

? mit_oldstyle

? mit_oldstyle_disclaimer

? mit_oldstyle_permission

? mpl

? mpl_1

? mpl_1_1

? mpl_2

? ms_pl

? ms_rl

? ntp

? ntp_disclaimer

? ofl

? ofl_1

? ofl_1_no_rfn

? ofl_1_rfn

? ofl_1_1

? ofl_1_1_no_rfn

? ofl_1_1_rfn

? openssl

? postgresql

? public_domain

? python

? python_2

? qpl

? rpsl

? rpsl_1

? sgi_b

? sgi_b_1

? sgi_b_1_1

? sgi_b_2

? ssleay

? unicode_strict

? unicode_tou

? wtfpl

? wtfpl_1

? wtfpl_2

? wtfnmfpl

? wtfnmfpl_1

? zlib

? zlib_acknowledgement

? zpl

? zpl_1

? zpl_1_1

? zpl_2

? zpl_2_1

Licensing traits

Patterns each covering a single trait occurring in licenses.

Each of these patterns has the tag `type:trait` .

? any_of

? clause_retention

? clause_reproduction

? clause_advertising

? clause_advertising_always

? clause_non_endorsement

? fsf_unlimited

? fsf_unlimited_retention

? license_label

? license_label_trove

? licensed_under

? or_at_option

? usage_rfn

? version

? version_later

? version_later_paragraph

? version_later_postfix

? version_number

? version_numberstring

? version_prefix

License combinations

Patterns each covering a combination of multiple licenses.

Each of these patterns has the tag `type:combo` .

? perl

License groups

Patterns each covering either of multiple licenses.

Each of these patterns has the tag `type:group` .

? bsd

? gnu

? mit

STRUCTURE

The regexp patterns follows the DefHash specification, and more specifically the structure of `Regexp::Pattern`, defining access to one pattern per DefHash object, as "pat".

Additionally, (sub)patterns are available in plaintext form, as "pat.alt.*".

SUBJECT

Each pattern targets one or more subjects, i.e. ways to directly or indirectly represent a license.

Beware that not all pattern objects fully cover all subjects.

trait

Distinguishing trait or feature expressed in licensing strings, e.g. an advertising clause or granting "...or any later version."

Coverage for this subject is currently rather weak.

name

Distinguishing name, e.g. usable in license grant.

grant

Text granting the license.

license

Text containing licensing terms.

Texts containing both license grant and licensing terms (e.g. BSD- and MIT-style licensing) may be classified as either subject grant or subject license.

This may change, as needs for distinction is better understood.

SCOPE

Each pattern can process material within some scope, i.e. a certain sample size of the full subject.

As an example, `<https://codesearch.debian.net/>` use line-based processing, where patterns spanning multiple lines are not applicable.

line

Pattern typically found within a single line. Typically this means less than 70 characters within scope.

sentence

Pattern matching within a sentence.

May span multiple lines, but not across multiple sentences. Typically this means no full-stop or colon within scope.

paragraph

Pattern matching distinguishing paragraph.

May span multiple sentences, but not multiple paragraphs. Typically this means no newline within scope.

section

Pattern matching distinguishing section.

May span multiple paragraphs, but not multiple sections. Typically this means blank line within scope.

multisection

Pattern may span multiple sections.

TAGS

Pattern defhashes optionally includes tags, which may help in selecting multiple related patterns.

Tags are hierarchical, with ":" as separator, and may be extended without notice.

Therefore take care to permit sub-parts when tag-matching, e.g. using a regex like

" ^\Asome:tag(?:\z:)/ ".

? family:bsd

? family:cc

? family:gpl

? family:mit

? family:zlib

Pattern covers a license part of a family of licenses.

? license:contains:grant

License mentions a preferred form for granting the license.

This implies that license is commonly granted by use of a different (typically far shorter) text.

? license:contains:license:*

License contains another license.

Wildcard is the key to corresponding license pattern.

? license:contains:name:*

License mentions name of another license.

? license:is:grant

License is commonly granted by stating the whole license.

? type:usage:*.*

Pattern covers a specific usage of a license.

Third part is the key of the corresponding non-usage-specific pattern.

Fourth part is the key of the corresponding usage trait pattern.

? type:combo

Pattern covers a combination of multiple licenses.

? type:group

Pattern covers either of multiple licenses.

? type:singleversion:*

Pattern covers a specific version of a license.

Last part of tag is the key of the corresponding non-version-specific pattern.

? type:trait

Pattern covers a single trait occurring in licenses.

? type:unversioned

Pattern covers a license without versioning scheme.

? type:versioned:decimal

Pattern covers a license using decimal number versioning scheme.

DEPRECATED TAGS

Tags not documented in this POD, specifically non-hierarchical tags, are deprecated and will be dropped in a future release.

AUTHOR

Jonas Smedegaard "<dr@jones.dk>"

COPYRIGHT AND LICENSE

Copyright ? 2016-2017 Jonas Smedegaard

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this

program. If not, see <<https://www.gnu.org/licenses/>>.

perl v5.30.0

2020-02-21

Regexp::Pattern::License(3pm)